

Storing data inside of images [v2.0]

Section I: File format

First of all, in order to retain data, you must ensure that the image file always meets these criteria:

- It has to be in 24 bit color (or higher, although it may become less efficient).
- It cannot be resized (You can make the image taller, but not wider, or smaller (see paragraph 4)).
- It must be stored in a lossless and/or non compressing format.

In 24 bit color images, each pixel has three values, R (red), G (green), and B (blue). Each of these values can hold any value between 0 and 255. That means that each value can store 8 bits of data. So, naturally, we split the file to encode into sections of 8 bits and store them in the image file. The data layout is as follows:

File Length (bytes) [32 bit]
File Data [Rest of file]

Pixels are read horizontally from left to right. Here is an example of a 4x4 image (holds 384 bytes) (with pixels numbered, the image would be read from the lowest numbered pixels, to the largest numbered pixels):

00	01	02	03
04	05	06	07
08	09	10	11
12	13	14	15

After the parser has read in the amount of data defined in "File Length" (the first 32 bits of the data), it will stop processing the file, and output the file data.

This means that the maximum file size is ~4095 GB (theoretically).

Section II: HTTP API

An HTTP API is also provided to enable retrieving data inside images hosted on free image hosting websites on the World Wide Web.

Basic steps to using the API:

1. Encode data into image
2. Upload image to free image hosting website
3. Ensure that the image still meets the specifications as listed above
4. Copy the direct URL of the image file (ex, <http://www.example.com/foo.png>)
5. Paste the link into the URL (<http://ig.acfsys.net/g?u=>), and modify as necessary (ex, add fn and ct attributes)

This is the API's URL, modify it as necessary:

[http://ig.acfsys.net/g?u=<Path to image>&fn=<filename \(for downloading\)>&ct=<MIME type>](http://ig.acfsys.net/g?u=<Path to image>&fn=<filename (for downloading)>&ct=<MIME type>)

An example image hosted on TinyPIC that contains the text "hello, world":

<http://i28.tinypic.com/2b8oex.png>

The corresponding API call would be:

<http://ig.acfsys.net/g?u=http://i28.tinypic.com/2b8oex.png&ct=text/html>

This would display the text "hello, world" in the user's browser. To trigger a file download box:

<http://ig.acfsys.net/g?u=http://i28.tinypic.com/2b8oex.png&fn=hello.txt&ct=application/octet-stream>

You can also use <http://ig.acfsys.net/g/filename.ext?u=...> to set the filename of the file for download.

This API can be used to host large or small files on image hosting services and make them available for users to download just by providing them with the appropriate URL (or a shortened version, such as that provided by tr.im)